

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of : THE COMMISSIONER IS AUTHORIZED
Kazutoshi FUNAHASHI et al. : TO CHARGE ANY DEFICIENCY IN THE
Serial No. NEW : FEES FOR THIS PAPER TO DEPOSIT
Filed March 18, 2004 : ACCOUNT NO. 23-0975
: **Attn: APPLICATION BRANCH**
: Attorney Docket No. 2004_0425A

DATA SHARING APPARATUS AND PROCESSOR
FOR SHARING DATA BETWEEN PROCESSORS
OF DIFFERENT ENDIANNESS

CLAIM OF PRIORITY UNDER 35 USC 119

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

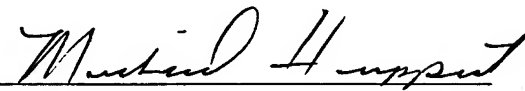
Sir:

Applicants in the above-entitled application hereby claim the date of priority under the International Convention of Japanese Patent Application No. 2003-075198, filed March 19, 2003, as acknowledged in the Declaration of this application.

A certified copy of said Japanese Patent Application is submitted herewith.

Respectfully submitted,

Kazutoshi FUNAHASHI et al.

By 
Michael S. Huppert
Registration No. 40,268
Attorney for Applicants

MSH/kjf
Washington, D.C. 20006-1021
Telephone (202) 721-8200
Facsimile (202) 721-8250
March 18, 2004



日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 3 月 1 9 日
Date of Application:

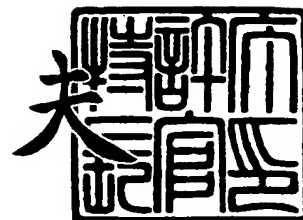
出 願 番 号 特 願 2 0 0 3 - 0 7 5 1 9 8
Application Number:
[ST. 10/C]: [J P 2 0 0 3 - 0 7 5 1 9 8]

出 願 人 松 下 電 器 産 業 株 式 会 社
Applicant(s):

2 0 0 4 年 1 月 2 1 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 5037740128

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/16

【発明者】

 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

 【氏名】 舟橋 和年

【発明者】

 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

 【氏名】 永安 勝

【発明者】

 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

 【氏名】 井川 智

【特許出願人】

 【識別番号】 000005821

 【氏名又は名称】 松下電器産業株式会社

【代理人】

 【識別番号】 100109210

 【弁理士】

 【氏名又は名称】 新居 広守

【手数料の表示】

 【予納台帳番号】 049515

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0213583

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 プロセッサ、データ共有装置

【特許請求の範囲】

【請求項 1】 データバスを介してメモリに接続されるプロセッサであって

、
データバスよりも小さい幅のデータについてメモリアクセスする場合に、データバスにおける当該データの位置を逆転させた位置を指すようにアドレスの下位ビットを変換して前記メモリに出力するアドレス変換手段を備え、

当該プロセッサのエンディアンとは逆のエンディアンによるバイトオーダで前記メモリとの間でデータが転送されるよう前記データバスに接続される

ことを特徴とするプロセッサ。

【請求項 2】 前記プロセッサとは異なるエンディアンの他のプロセッサとの間で前記メモリを介して共有すべき構造体データについて、前記他のプロセッサに実行されるプログラムにおける前記構造体データの定義に対して、基本語長より小さいデータを基本語長内で逆のオーダで定義したプログラムを実行する

ことを特徴とする請求項 1 記載のプロセッサ。

【請求項 3】 前記プロセッサは、さらに、

前記プロセッサのエンディアンによるバイトオーダでデータバスに接続されたキャッシュメモリを備え、

前記プロセッサは、データバスと同じ幅のデータ単位で前記メモリからキャッシュメモリにデータを読み込む

ことを特徴とする請求項 1 又は 2 記載のプロセッサ。

【請求項 4】 互いに異なるエンディアンの第 1 プロセッサ及び第 2 プロセッサと、メモリとを有するデータ共有装置であって、

第 1 プロセッサ及び第 2 プロセッサは、共に第 1 プロセッサのエンディアンによるバイトオーダでデータバスを介して前記メモリに接続される

ことを特徴とするデータ共有装置。

【請求項 5】 前記データ共有装置は、さらに、

第 2 プロセッサがデータバスよりも小さい幅のデータについてメモリにアクセ

スする場合に、データバスにおける当該データの位置を逆転させた位置を指すようにアドレスの下位ビットを変換して前記メモリに出力するアドレス変換手段を備えることを特徴とする請求項 4 記載のデータ共有装置。

【請求項 6】 前記メモリは、第 1 プロセッサ及び第 2 プロセッサにアクセスされる構造体データを記憶し、

前記第 1 プロセッサは、前記構造体データを定義した第 1 プログラムを実行し、

前記第 2 プロセッサは、前記構造体データについて基本語長より小さい幅のデータを基本語長内で第 1 プログラムとは逆のオーダで定義した第 2 プログラムを実行する

ことを特徴とする請求項 5 記載のデータ共有装置。

【請求項 7】 前記データ共有装置は、さらに、ダイレクトメモリアクセスによるデータ転送を制御する転送手段を備え、

前記転送手段は、転送元と転送先とで異なるエンディアン of データを必要とし、かつデータバスよりも小さい幅のデータを転送する場合に、転送元と転送先とで基本語長内における当該データの順序を逆転させる

ことを特徴とする請求項 5 又は 6 記載のデータ共有装置。

【請求項 8】 前記転送手段は、転送元と転送先とで異なるエンディアンのデータを必要とし、かつデータバスよりも小さい幅のデータを転送する場合に、データバスにおける当該データの位置を逆転させた位置を指すように、転送元及び転送先の方のアドレスの下位ビットを変換して前記メモリに出力する変換部を有する

ことを特徴とする請求項 7 記載のデータ共有装置。

【請求項 9】 さらに、

第 2 プロセッサのエンディアンによるバイトオーダでデータバスに接続されたキャッシュメモリを備える

ことを特徴とする請求項 4 記載のデータ共有装置。

【請求項 10】 前記データ共有装置は、さらに、

第 2 プロセッサがデータバスよりも小さい幅のデータについてメモリにアクセ

スする場合に、データベースにおける当該データの位置を逆転させた位置を指すようにアドレスの下位ビットを変換して前記メモリに出力するアドレス変換手段を備えることを特徴とする請求項 9 記載のデータ共有装置。

【請求項 11】 前記メモリは、第 1 プロセッサ及び第 2 プロセッサにアクセスされる構造体データを記憶し、

前記第 1 プロセッサは、前記構造体データを定義した第 1 プログラムを実行し、

前記第 2 プロセッサは、前記構造体データについて基本語長より小さい幅のデータを基本語長内で第 1 プログラムとは逆のオーダで定義した第 2 プログラムを実行する

ことを特徴とする請求項 10 記載のデータ共有装置。

【請求項 12】 前記データ共有装置は、さらに、ダイレクトメモリアクセスによるデータ転送を制御する転送手段を備え、

前記転送手段は、転送元と転送先とで異なるエンディアン of データを必要とし、かつデータベースよりも小さい幅のデータを転送する場合に、転送元と転送先とで基本語長内における当該データの順序を逆転させる

ことを特徴とする請求項 10 又は 11 記載のデータ共有装置。

【請求項 13】 前記転送手段は、転送元と転送先とで異なるエンディアンのデータを必要とし、かつデータベースよりも小さい幅のデータを転送する場合に、データベースにおける当該データの位置を逆転させた位置を指すように、転送元及び転送先 of 一方のアドレスの下位ビットを変換して前記メモリに出力する変換部を有する

ことを特徴とする請求項 12 記載のデータ共有装置。

【請求項 14】 互いに異なるエンディアンの第 1 プロセッサ及び第 2 プロセッサと、第 1 プロセッサ及び第 2 プロセッサとが共に第 1 プロセッサ of エンディアンによるバイトオーダでデータベースを介して接続されたメモリとを有するデータ処理装置におけるデータ共有方法であって、

前記メモリにて共有すべき構造体データについて、前記第 1 プロセッサにおける前記構造体データの定義に対して、基本語長より小さいデータを基本語長内

で逆のオーダで定義したプログラムを第2プロセッサに実行させるステップと、
第2プロセッサがデータバスよりも小さい幅のデータについてメモリにアクセスする場合に、データバスにおける当該データの位置を逆転させた位置を指すようにアドレスの下位ビットを変換する変換ステップと
を有することを特徴とするデータ共有方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 互いに異なるエンディアン（Endian）の2つのプロセッサとメモリとを有するデータ共有装置及びそのプロセッサに関し、特にプロセッサ間のデータ共有に関する。

【0002】

【従来の技術】

プロセッサの基本語長が2バイト（16ビット）以上である場合に、2バイト以上のデータをメモリに格納するバイトの順番として、いわゆるビッグエンディアン（Big Endian）とリトルエンディアン（Little Endian）との2つの方式がある。メモリがバイト単位にアドレスが割り当てられているのに対して、基本語長が2バイト以上あるからである。

【0003】

まず、ビッグエンディアンについて説明する。

ビッグエンディアンとは、バイト単位のアドレスを有するメモリに2バイト以上のデータを格納する際のバイト列の並び順（バイトオーダ）の方式であって、ビッグエンド（つまりMSB（Most Significant Bit）側。この場合Bは、BitではなくByteというべきである。）のバイトから、メモリのアドレス昇順に格納する方式をいう。ただし、エンディアンは、ビット列の並び順ではなくバイト列の並び順であって、バイト内のビットの並び順は変わらない。

【0004】

図14は、ビッグエンディアンの説明図を示す。同図のように、例えば4バイトデータ”89ABCDEF”（16進数、以下の” ” 内も同じ）をビッグエンディアンでアドレス100～103番地のメモリ領域に格納する場合、100～103番地の順に、”89”、” ” AB”、” CD”、” EF”と格納される。

【0005】

図15は、図14と同じ例であるが、32ビットを基本語長とした場合のメモリイメージ（メモリデータ）を示している。メモリアドレスはバイト単位であり、基本語長の中で、左から順に下位アドレス（アドレス下位2ビットによりアドレッシングされるバイト位置）が”0”、”1”、”2”、”3”と昇順になっている。最上位バイト”89”は0番地に格納される。

【0006】

このようにビッグエンディアンでは、MSB側のバイトからアドレス昇順にメモリに格納される。

図16は、ビッグエンディアンCPU2とメモリ3bとを接続する一般的な構成を示す。なお、本明細書中CPUは単にプロセッサとも呼ぶ。

【0007】

同図のようにCPU2のデータ入出力端子D[31:23]、D[24:16]、D[15:8]、D[7:0]は、データバスによってメモリ3aの下位アドレス0、1、2、3にそれぞれ接続される。これにより、メモリのアドレス昇順に、MSBからバイトデータが格納されることになる。また、同図のメモリ3bには、同図の構造体の例(structure sample)4が、CPU2内のプログラムで定義されていた場合の、メモリイメージを示している。LONG INTEGER wは、基本語長(32ビット)のワードデータであり、MSBから順にw3、w2、w1、w0の4バイトからなる。SHORT INTEGER xは、ハーフワードのデータであり、MSBから順にx1、x0からなる。SHORT INTEGER yも同様である。CHAR a、b、c、dはそれぞれバイトデータである。

【0008】

次に、リトルエンディアンについて説明する。

リトルエンディアンとは、リトルエンド（つまりLSB (Least Significant Bit) 側）のバイトからメモリのアドレス昇順に格納する方式をいう。

【0009】

図17は、リトルエンディアンの説明図を示す。同図は、図14と同じデータ例であり、32ビットデータ”89ABCDEF”は、100～103番地の

順に、“EF”、“CD”、“AB”、“89”と格納される。

【0010】

図18は、図17と同じ例について、32ビットを基本語長としたメモリイメージを示している。図15と比べて、下位アドレスの並びが逆であり、右から順に0、1、2、3となっている。

【0011】

図19は、リトルエンディアンCPU1とメモリ3aとを接続する一般的な構成を示す。同図のようにCPU1のデータ入出力端子D[7:0]、D[15:8]、D[24:16]、D[31:23]は、データバスによってメモリ3aの下位アドレス0、1、2、3にそれぞれ接続される。これにより、メモリのアドレス昇順に、LSB側のバイトからバイトデータが格納されることになる。また、同図のメモリ3aには、図16と同じ構造体の例(structure sample)4が、CPU1内のプログラムで定義されていた場合の、メモリイメージを示している。図16のメモリイメージと比較するとバイト列が逆になっている。

【0012】

このようにビッグエンディアン、リトルエンディアンは、2バイト以上のデータをメモリに格納する場合の、バイト列のオーダの方式であり、順序が逆になっている。

【0013】

このような相違から、ビッグエンディアンCPUとリトルエンディアンCPUとが混在するシステムでは、共有データの同一性を確保するためにエンディアンの違いを変換する構成が必要になる。

【0014】

特許文献1は、ビッグエンディアンCPUとリトルエンディアンCPUとの間でパケット通信を行う際に、ビッグエンディアン・プロセッサにおいてパケット上のデータをリトルエンディアンに変換するエンディアン変換部を備えるパケット通信装置を開示している。

【0015】

また、特許文献2は、アクセスアドレスとアクセスサイズとに基づいてストア

データを変換することによって、データのサイズ／アライン／エンディアンの違いを吸収してメモリアクセスを行うデータアライン装置を開示している。

【0016】

【特許文献1】

特開平06-69978

【0017】

【特許文献2】

特開2000-3304

【0018】

【発明が解決しようとする課題】

しかしながら、特許文献1、2に開示された技術では、エンディアン変換部を備えるプロセッサには特殊なハードウェアを付加しなければならない。また、エンディアン変換による遅延が発生するのでメモリアクセスの高速化の壁を作ってしまう、メモリデータを共有する場合に他方のプロセッサまで高速化の障害となるという問題を有している。

【0019】

また、一般に、通信はリトルエンディアンのプロセッサ、画像処理はビッグエンディアンのプロセッサで処理している。近年、携帯電話が画像処理機能を有するようになり、リトルエンディアンのプロセッサとビッグエンディアンのプロセッサとが同一バスに接続されるようになってきた。そして、リアルタイムにデータを共有する必要性が出てきた。

【0020】

本発明は、ビッグエンディアン・プロセッサとリトルエンディアン・プロセッサがバスに接続されている場合に、極めて簡易な構成でメモリデータの共有を実現するプロセッサ及びメモリ共有装置を提供することを目的とする。

【0021】

【課題を解決するための手段】

上記課題を解決するため、本発明のプロセッサは、データバスを介してメモリに接続されるプロセッサであって、データバスよりも小さい幅のデータについ

てメモリアクセスする場合に、データベースにおける当該データの位置を逆転させた位置を指すようにアドレスの下位ビットを変換して前記メモリに出力するアドレス変換手段を備え、当該プロセッサのエンディアンとは逆のエンディアンによるバイトオーダで前記メモリとの間でデータが転送されるよう前記データベースに接続されることを特徴とする。

【0022】

この構成によれば、データベースと同じ幅のデータについて、メモリに対して、第1プロセッサも第2プロセッサも同じバイトオーダでアクセスすることになるので、データベースと同じ幅のデータの共有を極めて簡単な構成で実現できるという効果がある。また、アドレス変換手段によって、プロセッサは、データベースよりも小さい幅のデータについて、メモリにアクセスすることが可能となる。さらに、アドレス変換手段による簡単なハードウェアによって構成できるので、メモリアクセスの高速化の障害とはならない。

【0023】

ここで、前記プロセッサ（第1プロセッサ）とは異なるエンディアンの他のプロセッサ（第2プロセッサ）との間で前記メモリを介して共有すべき構造体データについて、前記第2プロセッサに実行されるプログラムにおける前記構造体データの定義に対して、前記プロセッサは基本語長より小さいデータを基本語長内で逆のオーダで定義したプログラムを実行する構成としてもよい。

【0024】

この構成によれば、第1プロセッサと第2プロセッサとの間で、データベース幅よりも小さいデータに対しても、メモリを介してデータを共有することができる。例えば、データベース幅が32ビット(4バイト)である場合、アドレス変換手段を備えることによって、第1プロセッサから見たメモリの0番地と第2プロセッサから見たメモリの3番地とは同じメモリ領域を指す（同様に、第1プロセッサから見た1、2、3番地は、第2プロセッサから見た2、1、0番地である）。さらに第1プログラムにおいて構造体データがB0、B1、B2、B3という4バイトからなると定義されている場合、第2プログラムにおいて構造体データはB3、B2、B1、B0という4バイトからなると定義されている。B0と定義

されたバイトデータ B0 に対して、第1プロセッサは0番地をアクセスし、第2プロセッサは3番地をアクセスすることになる。このようにして、データバス幅と同じ幅のデータアクセスだけでなく、小さい幅のデータアクセスの場合でもデータを共有することができる。

【0025】

さらに、前記プロセッサは、前記プロセッサのエンディアンによるバイトオーダでデータバスに接続されたキャッシュメモリを備え、前記プロセッサは、データバスと同じ幅のデータ単位で前記メモリからキャッシュメモリにデータを読み込む構成としてもよい。

【0026】

この構成によれば、メモリからキャッシュメモリにデータバスと同じ幅のデータを読み込むので、エンディアンの異なるプロセッサのデータを極めて簡単な構成でキャッシュすることができるという効果がある。加えてキャッシュアクセスにアドレス変換が不要であり且つあるデータが共有メモリに置かれていてもキャッシュメモリに置かれていてもプロセッサは同じアクセス動作によってデータをアクセスすることができる。

【0027】

また、本発明のデータ共有装置は、互いに異なるエンディアンの第1プロセッサ及び第2プロセッサと、メモリとを有するデータ共有装置であって、第1プロセッサ及び第2プロセッサは、共に第1プロセッサのエンディアンによるバイトオーダでデータバスを介して前記メモリに接続されることを特徴とする。

【0028】

この構成によれば、データバスと同じ幅のデータについて、メモリに対して、第1プロセッサも第2プロセッサも同じバイトオーダでアクセスすることになるので、当該データの共有を極めて簡単な構成で実現できるという効果がある。

【0029】

ここで、前記データ共有装置は、さらに、第2プロセッサがデータバスよりも小さい幅のデータについてメモリにアクセスする場合に、データバスにおける当該データの位置を逆転させた位置を指すようにアドレスの下位ビットを変換して

前記メモリに出力するアドレス変換手段を備える構成としてもよい。

【0030】

この構成によれば、アドレス変換手段により、第2プロセッサは、データバスよりも小さい幅のデータについて、メモリにアクセスすることが可能となる。

ここで、前記メモリは、第1プロセッサ及び第2プロセッサにアクセスされる構造体データを記憶し、前記第1プロセッサは、前記構造体データを定義した第1プログラムを実行し、前記第2プロセッサは、前記構造体データについて基本語長より小さい幅のデータを基本語長内で第1プログラムとは逆のオーダで定義した第2プログラムを実行する構成としてもよい。

【0031】

この構成によれば、第1プロセッサと第2プロセッサとの間で、データバスと同じ幅のデータだけでなく、データバス幅よりも小さいデータに対しても、メモリを介してデータを共有することができる。また、アンキャッシュブルなデータ（キャッシュメモリに格納されないデータ）をエンディアンの異なるプロセッサ間で共有することが可能となる。

【0032】

ここで、前記データ共有装置は、さらに、ダイレクトメモリアクセス（DMA）によるデータ転送を制御する転送手段を備え、前記転送手段は、転送元と転送先とで異なるエンディアンのデータを必要とし、かつデータバスよりも小さい幅のデータを転送する場合に、転送元と転送先とで基本語長内における当該データの順序を逆転させる構成としてもよい。

【0033】

この構成によればエンディアンの異なる転送元と転送先との間でそれぞれに必要なデータの並び順を確保してDMA転送することができる。また、転送先でハーフワード単位及びバイト単位でのアクセスでもデータを利用することができる。

【0034】

ここで、前記転送手段は、転送元と転送先とで異なるエンディアンのデータを必要とし、かつデータバスよりも小さい幅のデータを転送する場合に、データバ

スにおける当該データの位置を逆転させた位置を指すように、転送元及び転送先の一方のアドレスの下位ビットを変換して前記メモリに出力する変換部を有する構成としてもよい。

【0035】

この構成によれば、変換部はアドレスの下位数ビットを変換するだけなので簡単な極めて簡単な構成により実現できる。

ここで、第2プロセッサのエンディアンによるバイトオーダでデータバスに接続されたキャッシュメモリを備える構成としてもよい。

【0036】

この構成によれば、データバスと同じ幅のデータについて、メモリからキャッシュメモリにデータを読み込むので、エンディアンの異なる第1プロセッサのデータを極めて簡単な構成でキャッシュできるという効果がある。

【0037】

また、本発明のデータ共有方法も上記データ共有装置と同様である。

【0038】

【発明の実施の形態】

（実施の形態1）

図1は、本実施の形態におけるデータ処理装置の構成を示すブロック図である。同図のようにデータ処理装置は、第1プロセッサ10、第2プロセッサ20、共有メモリ30、アドレス変換部21、キャッシュメモリ23を備え、データバス(図中のDバス)を介してこれらが接続されている。

【0039】

このデータ処理装置は、(A) ビッグエンディアンの第2プロセッサ20をリトルエンディアンの第1プロセッサ10と同じバイトオーダで共有メモリ30に接続したこと、(B) 第2プロセッサ20のアドレス変換、(C) 第1プロセッサ10、第2プロセッサ20それぞれにおける構造体の定義方法の3点により、メモリによるデータの共有を極めて簡単な構成で実現する。(A) 及び (B) はハードウェア的な構成であり、(C) はソフトウェア的な構成である。

【0040】

< A バス接続 >

図 1 において共有メモリ 30 及びキャッシュメモリ 23 中の 3、2、1、0 は、下位アドレス（ここではアドレスの下位 2 ビット）によりアドレッシングされるバイト位置を表している。

【0041】

第 1 プロセッサ 10 は、リトルエンディアン・プロセッサであり、バス接続点 41、42 を介して共有メモリ 30 に接続される。この接続は、リトルエンディアン・プロセッサの一般的なメモリ接続方法（図 19 参照）と同じであり、第 1 プロセッサ 10 はリトルエンディアンのバイトオーダで共有メモリ 30 に接続されている。つまり、第 1 プロセッサ 10 は、データ入出力端子のリトルエンド D[7:0] が共有メモリ 30 の 0 番地に接続されているので、メモリのアドレス昇順にリトルエンドからバイト列を格納する。

【0042】

第 2 プロセッサ 20 は、ビッグエンディアン・プロセッサであり、バス接続点 43、42 によって、共有メモリ 30 に接続される。この接続は上記（A）に関する。すなわち、この接続方法は、ビッグエンディアン・プロセッサの一般的なメモリ接続方法（図 16 参照）ではなく、あえてリトルエンディアン・プロセッサの一般的なメモリ接続方法（図 19 参照）と同じにしてある。この接続において、第 2 プロセッサ 20 のデータ入出力端子は、ビッグエンドではなくリトルエンドのバイト D[7:0] から共有メモリ 30 のアドレス昇順に接続されることになる。別言すれば、バス接続点 43、42 によって、第 2 プロセッサ 20 のバイトオーダがビッグエンディアンからリトルエンディアンに入れ替えられている。

【0043】

この接続によれば、第 1 プロセッサ 10 と第 2 プロセッサ 20 は、共有メモリ 30 に対する 32 ビットデータ（以下ワードと呼ぶ）のアクセスによるデータ共有が可能となる。加えて、キャッシュメモリ 23 へ共有メモリ 30 から 32 ビット単位でデータをキャッシュする（読み込む）ことが可能となる。ただし、16 ビットデータ（ハーフワードと呼ぶ）、8 ビットデータ（バイト）のアクセスによるデータ共有は、この接続だけでは実現できない。これは、上記（B）（C）

とあいまって実現することができる。

【0044】

共有メモリ30は、その第1プロセッサ10のデータ入出力端子D[31:23]、D[24:16]、D[15:8]、D[7:0]が、下位アドレス（アドレスの下位2ビット（A1、A0））が指すバイト位置である3、2、1、0番地に対応している。

【0045】

<B アドレス変換>

アドレス変換部21は、第2プロセッサ20の下位アドレスを変換して共有メモリ30に出力する。このアドレス変換は上記（B）に関し、基本語長である32ビットより小さいデータについて、基本語長内の当該データの位置を逆転させた位置を指すように下位アドレスを変換する。

【0046】

図2は、アドレス変換部21によるアドレス変換の入出力論理を示す図である。アドレス変換部21への入力は、「アクセスサイズ」、「変換前アドレス」であり、何れも第2プロセッサ20から入力される。アクセスサイズは、ワードアクセス、ハーフワードアクセス、バイトアクセスの何れかを示す。変換前アドレスは、アドレスバスの最下位から2ビットつまりA1、A0である。

【0047】

アドレス変換部21の出力は変換後アドレスである。変換後アドレスについて、同図に示すように、アドレス変換部21は、ワードアクセスの場合は、A1、A0をそのまま出力する（つまり、無変換である）。ハーフワードアクセスの場合は、A1のみ反転しA0をそのまま出力する。バイトアクセスの場合はA1とA0を共に反転して出力する。ここでは、ワードアクセス、ハーフワードアクセスにおいて、第2プロセッサ20はミスアライメントのデータにはアクセスしないものとする。

【0048】

図3は、第2プロセッサ20がワードをアクセスする場合の説明図である。第2プロセッサ20がワード単位でアクセスする場合は、共有メモリ30のデータを第1プロセッサ10と共有することができる。この場合、アドレス変換部21

はアドレス変換しない。

【0049】

図4(a)は、第2プロセッサ20がハーフワードをアクセスする場合の説明図である。ハーフワードアクセスではアドレス変換部21によって、変換前の下位アドレス0番地(A1、A0=0、0)は下位アドレス2番地(1、0)に、また下位アドレス2番地(A1、A0=1、0)は下位アドレス0番地(0、0)に変換されることになる。同図(a)のように、第2プロセッサ20がメモリの0番地にハーフワード"89AB"をストアする命令(movhw #89ABh, mem(0))を実行すると、メモリエイメージ(メモリデータ)4bのように2番地と3番地に"89AB"をストアすることになる。これに対して、第1プロセッサ10が同様の命令を実行すると、メモリエイメージ4aのように0番地と1番地に"89AB"をストアすることになる。

【0050】

このように、ハーフワードアクセスにおいて、第1プロセッサ10から見た共有メモリ30の下位アドレス0番地は、第2プロセッサ20から見た共有メモリ30の下位アドレス2番地であり、同様に、第1プロセッサ10から見た2番地は、第2プロセッサ20から見た0番地となっている。

【0051】

図5(a)は、第2プロセッサ20がバイトアクセスする場合の説明図である。バイトアクセスではアドレス変換部21によって、例えば変換前の下位アドレス0番地(A1、A0=0、0)はアドレス3番地(1、1)に変換されることになる。同図(a)のように、第2プロセッサ20がメモリの0番地にバイト"89"をストアする命令(movb #89h, mem(0))を実行すると、メモリエイメージ5bのように3番地に"89"をストアすることになる。これに対して、第1プロセッサ10が同様の命令を実行すると、メモリエイメージ5aのように0番地に"89"をストアすることになる。

【0052】

このように、バイトアクセスでは、第2プロセッサ20は、アドレス変換部21によって、下位アドレス0番地は3番地に、1番地は2番地に、2番地は1番

地に、3番地は0番地に変換されることになる。つまり、バイトアクセスにおいて、第1プロセッサ10から見た共有メモリ30の下位アドレス0、1、2、3番地は、第2プロセッサ20から見た共有メモリ30の下位アドレス3、2、1、0番地となっている。

【0053】

ところで、上記(A)に(B)のアドレス変換を加えても、第2プロセッサ20は共有メモリ30を介して第1プロセッサ10との間でハーフワード単位及びバイト単位でデータを共有することまでは実現されていない。(B)のアドレス変換は、(A)によるバイトオーダの入れ替えに伴って、第2プロセッサ20がハーフワード及びバイトにてアクセスできるようにした点に意味がある。

【0054】

つまり、アドレス変換部21を備えていなければ、第2プロセッサ20は共有メモリ30に対して、ワード単位のアクセスが可能であるものの、ハーフワード及びバイト単位でのアクセスができなくなる。例えば、図4(a)の例では、共有メモリ30の0、1番地は、メモリイメージ4aのようにワードの下位側(図の右側)ハーフワードであるが、第2プロセッサ20は0、1番地のつもりでワードの上位側のハーフワードに"89AB"を出力している。この状態で、仮にアドレス変換なしで0番地を共有メモリ30に指定すれば、共有メモリ30において"89AB"が入力されていない下位側ハーフワード(0、1番地)に、バス上の無効なデータを書き込んでしまうことになる。

【0055】

このように(B)のアドレス変換は、(A)のバイトオーダ入れ替えによって生じるハーフワード、バイトアクセスの不具合を解消する点に意味がある。また、アドレス変換部21は、簡単な組み合わせ回路で実現できるので、変換による下位アドレス出力の遅延は無視できる。

【0056】

また、ハーフワード、バイトに関して、図4(a)、図5(a)のように第1プロセッサ10から見たアドレスと、第2プロセッサ20から見たアドレスとが食い違っているから、ハーフワード単位、バイト単位のデータ共有はできないよ

うに思える。この食い違いに対しては、例えば、図4(b)、図5(b)に示すように、第1プロセッサ10と第2プロセッサ20とが、ハーフワード及びバイトの共有データに対しては、食い違ったアドレスを直接指定すれば、ワード単位だけでなくハーフワード単位でもバイト単位でも共有可能となる。本実施形態では、この点を上記(C)により実現する。

【0057】

キャッシュメモリ23は、第2プロセッサ20に対してバス接続点44介して、一般的なビッグエンディアン・プロセッサとメモリとの接続方法(図16参照)により接続されている。図1では、キャッシュメモリ23は第2プロセッサ20外部に備えているが、第2プロセッサ20内部であってもよい。キャッシュメモリ23は、ワード単位で共有メモリ30のデータをキャッシュすることができる。これは上記(A)のバイトオーダーを入れ替える接続による。

【0058】

図6は、共有メモリ30及びキャッシュメモリ23のメモリイメージを示す図である。キャッシュメモリ23は、共有メモリ30からワード単位にデータをキャッシュすることにより、共有メモリ30と下位アドレスのオーダーが異なる点を除いて同一のメモリイメージを保持することができる。ただし、キャッシュメモリ23が、ハーフワード単位、バイト単位に共有メモリ30からキャッシュすることは、上記(A)の接続だけでは実現することができない。

【0059】

<C 構造体定義>

図7は、構造体の定義(上記(C))によりハーフワード単位、バイト単位のデータ共有を行う説明図である。同図において、第1プロセッサ10に実行される第1プログラム10aは、構造体10bのように定義されたソースプログラムから生成されている。また、第2プロセッサ20に実行される第2プログラム20aは、構造体20bのように定義されたソースプログラムから生成されている。構造体20bは、構造体10bと同じ変数を定義しているが、基本語長の範囲内では基本語長より小さい変数の順序を逆順に定義している。

【0060】

例えば、16ビット整数型の変数 x 、 y について構造体 10b では「SHORT INTEGER y, x ;」と定義され、構造体 20b では「SHORT INTEGER x, y ;」と逆順に定義されている。また、文字型の変数 a, b, c, d についても構造体 10b では「CHAR d, c, b, a ;」と定義され、構造体 20b では「CHAR a, b, c, d ;」と逆順に定義されている。これらの構造体はプログラマにより記述され、あるいはコンパイラによって逆順に記述するようにしてもよい。

【0061】

これにより、図4(a)、図5(a)に示したようなハーフワード及びバイトアクセスでのアドレスの食い違いを、構造体での定義の中で入れ替えることにより、許容することができる。つまり、図4(b)、図5(b)に示したように、共有すべき同一の変数に対して、第1プロセッサ10及び第2プロセッサ20がハーフワード及びバイトアクセスをする場合に、食い違ったアドレスを指すようにすることができる。

【0062】

図7に示すように、構造体10bによって定義された共有メモリ30のメモリイメージと、構造体20bによって定義されたメモリイメージとは一致する。ただし、第2プロセッサ20から見たハーフワード、バイトの下位アドレスは、第1プロセッサ10とは逆順になる(食い違っている)。例えば、変数 a について、第1プロセッサ10からは下位アドレス3番地に格納されていると見え、第2プロセッサ20内部では下位アドレス0番地と見える。これにより、バイトアクセスによっても変数 a を共有することができる。

【0063】

また、変数 y は、第1プロセッサ10からは下位アドレス0、1番地に格納されていると見え、第2プロセッサ20内部では下位アドレス2、3番地と見える。これにより、ハーフワードアクセスによっても変数 y を共有することができる。

【0064】

また、キャッシュメモリ23についても、共有すべき構造体として定義されたデータであれば、ワード単位だけでなくハーフワード単位でもバイト単位でもキ

ヤッシュすることができる。

【0065】

図8は、構造体の他の定義例を示す図である。同図において第1プロセッサ10用の構造体10cにおいて、最後の符号なし8ビット変数「CHAR b2;」の定義により基本語長内で8ビット余ることになる。そのため、構造体20cでは、余りと同じビット長のダミーバイトデータ「CHAR bDummy0;」を挿入している。これにより、構造体20cでは、8ビットの余りも含めて基本語長内で逆順に定義している。

【0066】

図9は、コンパイルの前処理として、ソースプログラム中の構造体の定義を逆順にする処理を示す図である。

同図のように、前処理部100は、第1プロセッサ10用の第1ソースプログラムと第2プロセッサ20用の第2ソースプログラムを入力し、両者における共有すべき構造体を検出し（S11）、一方のソースプログラムにおける基本語長内の変数の順序が他方のソースプログラムにおける変数の順序と逆になるように変数の順序を入れ替える（S12）。その際、必要に応じて図8に示したダミーバイトデータ又はダミーハーフワードを挿入する。この後、両ソースプログラムはそれぞれのコンパイラにてコンパイルされる。

【0067】

これにより、プログラマが共有すべき構造体データの順序をいちいち考えてプログラムする必要がなくなり、共有すべき変数名だけを一致させておけばよいので、プログラマの負担を小さくすることができる。

【0068】

以上説明してきたように、本実施の形態におけるデータ処理装置によれば、（A）ビッグエンディアンの第2プロセッサ20をリトルエンディアンの第1プロセッサ10と同じバイトオーダで共有メモリ30に接続したこと（図1のバス接続点42、43参照）によって、リトルエンディアンの第1プロセッサ10とビッグエンディアンの第2プロセッサ20とで共有メモリをワード単位のアクセスによってデータを共有することができる。加えて、ビッグエンディアン・プロセ

ッサのキャッシュメモリに、リトルエンディアンの共有メモリからワード単位でキャッシュすることができる。

【0069】

また (A) に加えて (B) アドレス変換部 21 によって、第 2 プロセッサ 20 は、ハーフワード単位及びバイト単位でのアクセスによるデータ共有までは必ずしもできないが、リトルエンディアン用のメモリをアクセスすることができる。例えば、リトルエンディアンで接続された第 1 プロセッサ 10 と共有メモリ 30 に対して、ビッグエンディアンの第 2 プロセッサ 20 を (A) (B) という簡単な構成で、第 2 プロセッサ 20 のメモリとしてアクセスすることができる。

【0070】

さらに、(A)、(B) に加えて (C) 第 1 プロセッサ 10 と第 2 プロセッサ 20 とで構造体定義を基本語長内で基本語長より小さい変数を逆順に定義することによって、リトルエンディアンの第 1 プロセッサ 10 とビッグエンディアンの第 2 プロセッサ 20 とで共有メモリをワード単位のアクセスだけでなく、ハーフワード単位でもバイト単位でもデータを共有することができる。

【0071】

なお、上記実施の形態では、第 1 プロセッサ 10 がリトルエンディアン、第 2 プロセッサ 20 がビッグエンディアンである場合の構成を説明したが、逆のエンディアンであってもよい。その場合、共有メモリは両プロセッサからビッグエンディアンとして接続され、アドレス変換部はリトルエンディアン・プロセッサのアドレスを変換する構成とすればよい。

【0072】

また、上記実施の形態では、第 1 プロセッサ 10 及び第 2 プロセッサ 20 は 32 ビットを基本語長としているが、64 ビットや 128 ビットなど他の長さの基本語長であってもよい。この場合、アドレス変換部 21 は、基本語長の内部のバイト位置を示す下位アドレスを変換する構成とすればよい。

【0073】

(実施の形態 2)

上記実施の形態では、上記 (A) (B) によって共有メモリ 30 に格納される

データについて、第1プロセッサ10から見たときと第2プロセッサ20から見たときとでハーフワードアドレス及びバイトアドレスに食い違いが生じている点を上記(C)によって解決している。本実施の形態では、この食い違いを(C)によらないでDMA転送において解決する構成について説明する。

【0074】

図10は、本発明の実施の形態2におけるデータ処理装置の構成を示すブロック図である。同図のデータ処理装置は、図1と比較して、新たにSDRAM (Synchronous Dynamic Random Access Memory) 31、DMAC (Direct Memory Access Controller) 32、I/O (Input/Output device) 33、I/O 34が追加された点が異なっている。図1と同じ構成要素には同じ符号を付してあるので説明を省略し、異なる点を中心に説明する。

【0075】

本データ処理装置は、DVDレコーダやデジタル放送受信機などAV (Audio Visual) 機器に備えられる。この場合、第2プロセッサ20はMPEG (Moving Picture Expert Group) による圧縮動画像データの復号や符号化などの画像処理を行い、第1プロセッサ10は画像処理装置全体の制御を行う。共有メモリ30は、第1プロセッサ10と第2プロセッサ20との間の各種データの授受に用いられる。

【0076】

SDRAM31は、第1プロセッサ10及び第2プロセッサ20によって作業メモリとして利用され、MPEG圧縮動画像を示すストリームデータ、復号中のピクチャ、復号後の参照ピクチャ、復号後のオーディオデータなどを保持する。

【0077】

DMAC32は、DMA転送に際して、転送元データと転送先データとが、異なるエンディアンのデータであって、転送データサイズがワードよりも小さい場合に、バイトオーダを入れ替えるためのアドレス変換を行う。また、DMA転送は、メモリーメモリー間及びメモリーI/O間で行われる。ここで、メモリは共有メモリ30及びSDRAM31であり、I/OはI/O33、I/O34などである。

【0078】

I/O 33、34は、それぞれ8ビット、16ビットのI/Oデバイスであり、例えば、復号後のピクチャデータをシリアルビデオデータとして出力するビデオ出力部、復号後のオーディオデータを入力し音声信号として出力するオーディオ出力部などである。

【0079】

図11は、DMAC 32の詳細な構成を示すブロック図である。同図のようにDMAC 32は、第1アドレスカウンタ101、第2アドレスカウンタ102、ターミナルカウンタ103、データラッチ105及び制御部106を備える。

【0080】

第1アドレスカウンタ101、第2アドレスカウンタ102は、それぞれ転送元、転送先アドレスを保持し、転送する毎にカウントアップ又はカウントダウンする。両アドレスカウンタと転送元と転送先との対応付けは逆でもよい。

【0081】

ターミナルカウンタ103は、バースト転送時に転送すべきデータ数をカウントし終えると、制御部106に通知しDMA転送を終了させる。

アドレス変換部104は、転送元データと転送先データの間で、又はI/O 33、34とメモリデータの間で、エンディアンが異なりかつ転送データサイズがワードよりも小さい場合に、転送元と転送先とで基本語長内における当該データの順序を逆転させるために第1アドレスカウンタ101から出力される転送元／転送先アドレスを変換する。

【0082】

ここで、「エンディアンが異なる」というのは、転送元と転送先とが異なるエンディアンのプロセッサにアクセスされること、または、異なるエンディアンのプロセッサとI/Oに使用されることをいう。例えば、転送元データが第2プロセッサ20によりロード／ストアされ、転送先データが第1プロセッサ10によりロード／ストアされる場合や、転送元データが第2プロセッサ20によりストアされ、転送先がLSB側から順にバイトデータを要求するI/Oである場合等である。

【0083】

また、第2プロセッサによりロードストアされるデータはリトルエンディアンでメモリに格納されているので、ワード単位のDMAでは、アドレス変換部104によるアドレス変換なしでそのままDMA転送することができる。ところが、ワードより小さいハーフワード単位、バイト単位のDMAでは、図4(a)、図5(a)に示したように、第1プロセッサ10から見たアドレスと第2プロセッサ20から見たアドレスが食い違っている。そのためアドレス変換部104は、転送元と転送先が異なるエンディアンのデータを必要とする場合、ハーフワード単位及びバイト単位でのアドレスの食い違いを正すためにアドレスを変換する。

【0084】

図12は、アドレス変換部104によるアドレス変換の入出力論理を示す説明図である。アドレス変換部104への入力は、「エンディアン」「アクセスサイズ」「変換前アドレス」である。「エンディアン」は制御部106から入力され、転送元データと転送先データとが同じエンディアンであるか異なるエンディアンであるかを示す。より正確には、転送元データをアクセスするプロセッサと転送先データをアクセスするプロセッサとが同じエンディアンであるか異なるエンディアンであるかを示す。また、転送元、転送先の一方がI/Oであって、当該I/Oが必要とする転送データのバイトオーダ（又はハーフワードのオーダ）がリトルエンディアンである場合は、転送元／転送先データをアクセスするプロセッサが第1プロセッサ10であればエンディアンが「同じ」、第2プロセッサ20であればエンディアンが「異なる」となる。「アクセスサイズ」は、制御部106から入力され、ワード、ハーフワード、バイトの何れかを指す。「変換前アドレス」は、第1アドレスカウンタ101から入力されるアドレスの下位2ビットである。

【0085】

「変換後アドレス」に示すように、アドレス変換部104は、エンディアンが同じ場合は、ワード転送であってもハーフワード転送であってもバイト転送であっても、変換前アドレスをそのまま出力する(変換しない)。また、エンディアンが異なっても、ワード転送である場合には、変換前アドレスをそのまま出力

する（変換しない）。エンディアンが異なっていて、ハーフワード転送である場合には、A1のみを反転して出力する。エンディアンが異なっていて、バイト転送である場合には、A1、A0の両ビットを反転して出力する。

【0086】

このように、図12におけるエンディアンが異なる場合の入出力論理は、図2に示したアドレス変換部21の入出力論理と同じであるが、重要なのは、アドレス変換部21によって第2プロセッサ20のハーフワードアドレス及びバイトアドレスが第1プロセッサ10と食い違っていることに対して、食い違いを正すように変換している点である。

【0087】

データラッチ105は、メモリーメモリー間のDMA転送において転送データを一時的に保持する。

制御部106は、第1プロセッサ10又は第2プロセッサ20によってアクセス可能な制御レジスタを内部に有し、制御レジスタの設定内容に従ってDMAC32全体を制御する。制御レジスタは、転送元及び転送先のスタートアドレス（I/Oの場合はそのアドレス又は制御部106から出力されるI/Oセレクト信号の指定）、転送データサイズ、転送元及び転送先のエンディアン（第1プロセッサ10にアクセスされるデータか第2プロセッサ20にアクセスされるデータか）などを保持する。

【0088】

図13（a）は、DMAC32によるDMA転送の様子を示すタイムチャートの一例である。同図において、ソースメモリ（転送元メモリ）は、例えばSDRAM31、デスティネーションメモリ（転送先メモリ）は共有メモリ30である。SDRAM31の100番地以降に第2プロセッサ20によってストアされた一定量のデータを、共有メモリ30の900番地以降にDMA転送する場合の例を示している。また、SAは第1アドレスカウンタ101からアドレス変換部104を介して出力されるソースアドレスを、SDは転送元メモリから読み出されたハーフワードのソースデータを、DAは第2アドレスカウンタ102によるデスティネーションアドレスを、DDは転送先メモリに書き込まれるハーフワード

のデスティネーションデータを示す。

【0089】

この場合、第1アドレスカウンタ101は100番地をスタートアドレスとして2ずつインクリメントし、第2アドレスカウンタ102は900番地をスタートアドレスとして2ずつインクリメントする。アドレス変換部104は、エンディアン異なるハーフワードのDMA転送なので、A1のみを反転する。その結果、アドレスバスに出力されるソースアドレスSAは、102、100、106、104、・・・と出力され、デスティネーションアドレスDAは、900、902、904、906、・・・と出力される。これにより、転送元データは、リトルエンドのハーフワードから転送されることになる。

【0090】

図13(b)は、同図(a)によるDMA転送の転送元メモリと転送先メモリのメモリイメージを示す図である。同図のように、転送元メモリの下位アドレス0番地のハーフワードは転送先メモリの下位アドレス2番地に転送され、2番地のハーフワードは0番地に転送される。

【0091】

これにより、ビッグエンディアンの第2プロセッサ20によって格納された転送元データは、DMA転送を経た後、転送先メモリにおいて第1プロセッサ10によってハーフワード単位にアクセスすることができる。このように、DMAC32は、図4(a)図5(a)に示したアドレスの食い違いを正してDMA転送を行う。転送後のデータは、第1プロセッサ10によってハーフワード単位でアドレスが食い違うことなくアクセスすることができる。また、転送先の900番地がI/O34であるとする、DMA転送によりLSBから順にハーフワードを供給することができる。図13では、ハーフワードのDMA転送を示したがバイト転送の場合も同様である。

【0092】

また、メモリーI/O間のDMA転送の場合は、制御部106がI/Oに対して直接チップセレクト信号を出力するので、データラッチ105を介することなくメモリから読み出されたデータを直接I/Oに(又はこの逆方向に)転送するこ

とになる。その場合も、例えばビッグエンディアンの第2プロセッサ20によって格納されたデータをバイト単位又はハーフワード単位に、リトルエンドからI/Oに転送することができる。エンディアンが逆の場合も同様である。

【0093】

以上説明してきたように、本実施の形態におけるデータ処理装置によれば、上記の(C)構造体データとして定義されていないデータであっても、DMA転送の際に、アドレス変換部104によって上記(A)(B)によって生じるハーフワードアドレス及びバイトアドレスの食い違いを正すので、転送後のデータはハーフワード又はバイト単位のアクセスにより他方のプロセッサが正しく読み出すことができる。また、メモリーI/O間のDMA転送についても同様に、アドレス変換部104によってハーフワードアドレス及びバイトアドレスの食い違いを正すことができる。I/OがLSB側からデータを必要とする場合も、MSB側からデータを必要とする場合も、DMAC32により必要とされるデータを転送することができる。

【0094】

なお、アドレス変換部104は、第1アドレスカウンタ101から出力されるアドレスを変換する代わりに、第2アドレスカウンタ102から出力されるアドレスを変換するようにしてもよい。

【0095】

また、図11に示したDMACでは、ハーフワード又はバイト転送時にアドレス変換により送信先データにおいてハーフワード又はバイトの並びを入れ替えているが、転送元データをワード単位でデータラッチ105に取り込み、データラッチ105においてハーフワード又はバイトの並びを入れ替えて、転送先メモリにワード単位で書き込む構成としてもよい。

【0096】

【発明の効果】

本発明のデータ処理装置によれば、(A)ビッグエンディアンの第2プロセッサをリトルエンディアンの第1プロセッサと同じバイトオーダで共有メモリ30に接続したこと(図1のバス接続点42、43参照)によって、リトルエンディ

アン第1プロセッサとビッグエンディアンの第2プロセッサとで共有メモリをワード単位のアクセスによってデータを共有することができる。加えて、ビッグエンディアン・プロセッサのキャッシュメモリに、共有メモリからワード単位でキャッシュすることができる。

【0097】

また(A)に加えて(B)アドレス変換することによって、第2プロセッサは、ハーフワード単位及びバイト単位でのアクセスによるデータ共有までは必ずしもできないが、リトルエンディアン用のメモリをアクセスすることができる。

【0098】

さらに、(A)、(B)に加えて(C)第1プロセッサと第2プロセッサとでアプリケーションプログラム内での構造体定義を基本語長内で基本語長より小さい変数を逆順に定義することによって、リトルエンディアンの第1プロセッサとビッグエンディアンの第2プロセッサとで共有メモリをワード単位のアクセスだけでなく、ハーフワード単位でもバイト単位でもデータを共有することができる。

【0099】

さらに、(A)、(B)に加えて(D)DMA転送の際に(B)と同じアドレス変換を行うことにより、エンディアンの異なるデータをDMA転送し、転送元と転送先とでハーフワード単位及びバイト単位でのアクセスによるデータ共有を行うことができる。

【図面の簡単な説明】

【図1】

実施の形態1におけるデータ処理装置の構成を示すブロック図である

【図2】

アドレス変換部21によるアドレス変換の入出力論理を示す説明図である。

【図3】

第1プロセッサ10及び第2プロセッサ20が共有メモリ30をワードでアクセスする場合の説明図である。

【図4】

(a) 第1プロセッサ10及び第2プロセッサ20が共有メモリ30を同じアドレスでハーフワードをアクセスする場合の説明図である。

(b) 第1プロセッサ10及び第2プロセッサ20が共有メモリ30を異なるアドレスで同じハーフワードをアクセスする場合の説明図である。

【図5】

(a) 第1プロセッサ10及び第2プロセッサ20が共有メモリ30を同じアドレスでバイトデータをアクセスする場合の説明図である。

(b) 第1プロセッサ10及び第2プロセッサ20が共有メモリ30を異なるアドレスで同じバイトデータをアクセスする場合の説明図である。

【図6】

共有メモリ30及びキャッシュメモリ23のメモリエイムを示す図である。

【図7】

構造体の定義によりハーフワード単位、バイト単位のデータ共有を行う説明図である。

【図8】

構造体の他の定義例を示す図である。

【図9】

コンパイルの前処理として、ソースプログラム中の構造体の定義を逆順にする処理を示す図である。

【図10】

本発明の実施の形態2におけるデータ処理装置の構成を示すブロック図である。

【図11】

DMA C 3 2の詳細な構成を示すブロック図である。

【図12】

アドレス変換部104によるアドレス変換の入出力論理を示す説明図である。

【図13】

(a) DMA C 3 2によるDMA転送の様子を示すタイムチャートの一例である。

(b) 転送元メモリと転送先メモリのメモリエイメージを示す図である。

【図 14】

ビッグエンディアンの説明図を示す。

【図 15】

ビッグエンディアンの説明図を示す。

【図 16】

ビッグエンディアンCPUとメモリとを接続する一般的な構成を示す。

【図 17】

リトルエンディアンの説明図を示す。

【図 18】

リトルエンディアンの説明図を示す。

【図 19】

リトルエンディアンCPUとメモリとを接続する一般的な構成を示す。

【符号の説明】

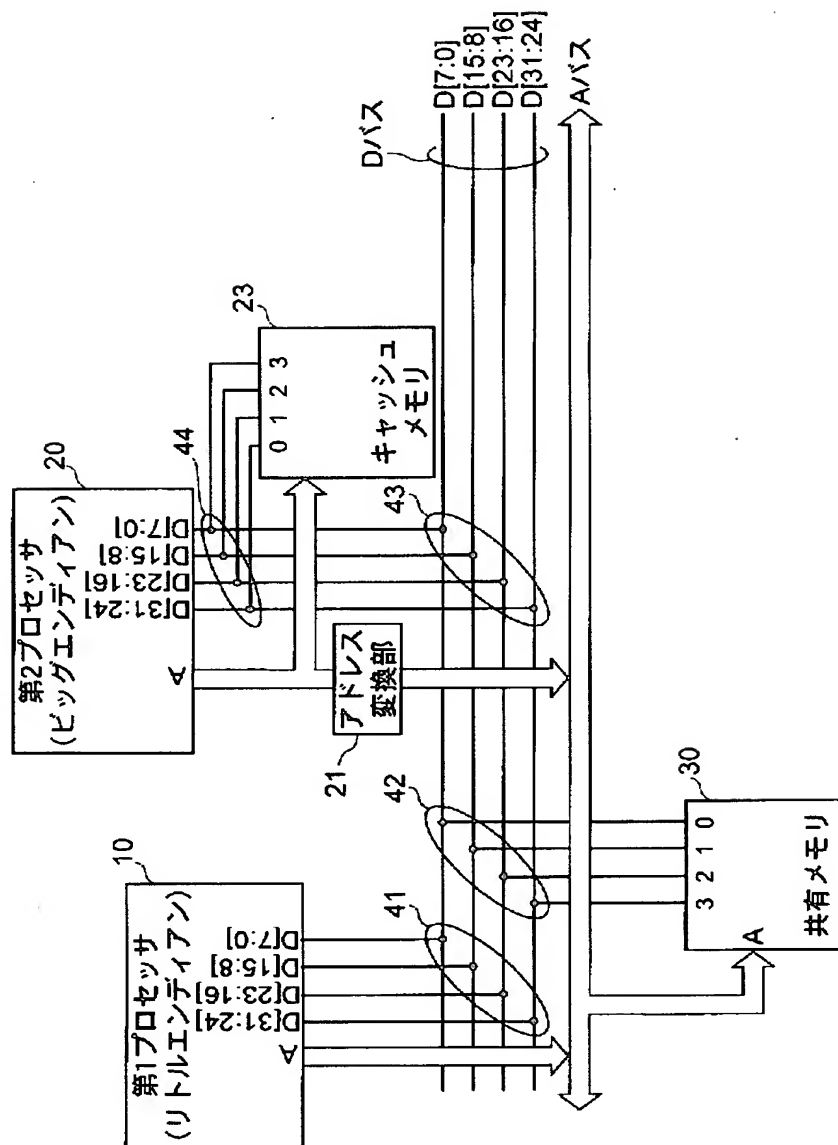
- 10 第1プロセッサ
- 10a 第1プログラム
- 10b、10c、20b、20c 構造体
- 20 第2プロセッサ
- 20a 第2プログラム
- 21 アドレス変換部
- 23 キャッシュメモリ
- 30 共有メモリ
- 31 SDRAM
- 32 DMAC
- 33 I/O
- 34 I/O
- 41 バス接続点
- 42 バス接続点
- 43 バス接続点

4 4	バス接続点
1 0 0	前処理部
1 0 1	アドレスカウンタ
1 0 2	アドレスカウンタ
1 0 3	ターミナルカウンタ
1 0 4	アドレス変換部
1 0 5	データラッチ
1 0 6	制御部

【書類名】

図面

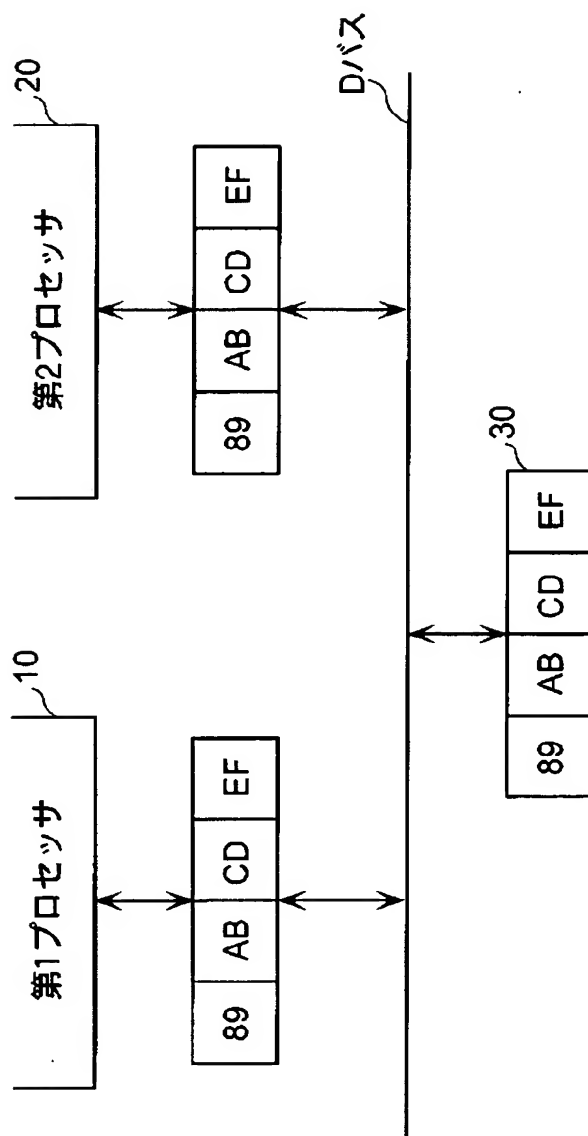
【図 1】



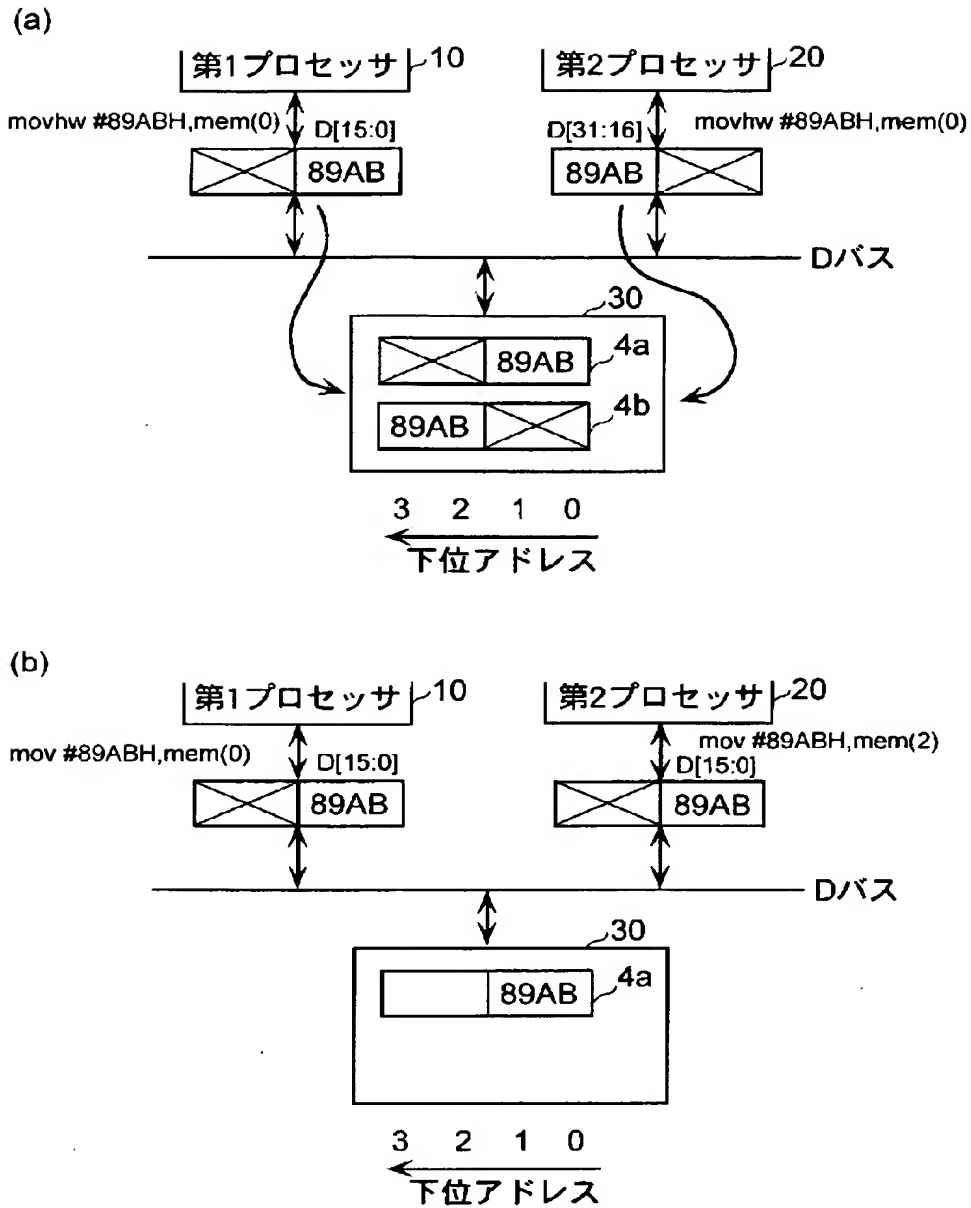
【図 2】

入力		出力
アクセスサイズ	変換前 アドレス	変換後 アドレス
w (32ビット)	A1, A0	A1, A0
hw (16ビット)	A1, A0	$\overline{A1}$, A0
B (8ビット)	A1, A0	$\overline{A1}$, $\overline{A0}$

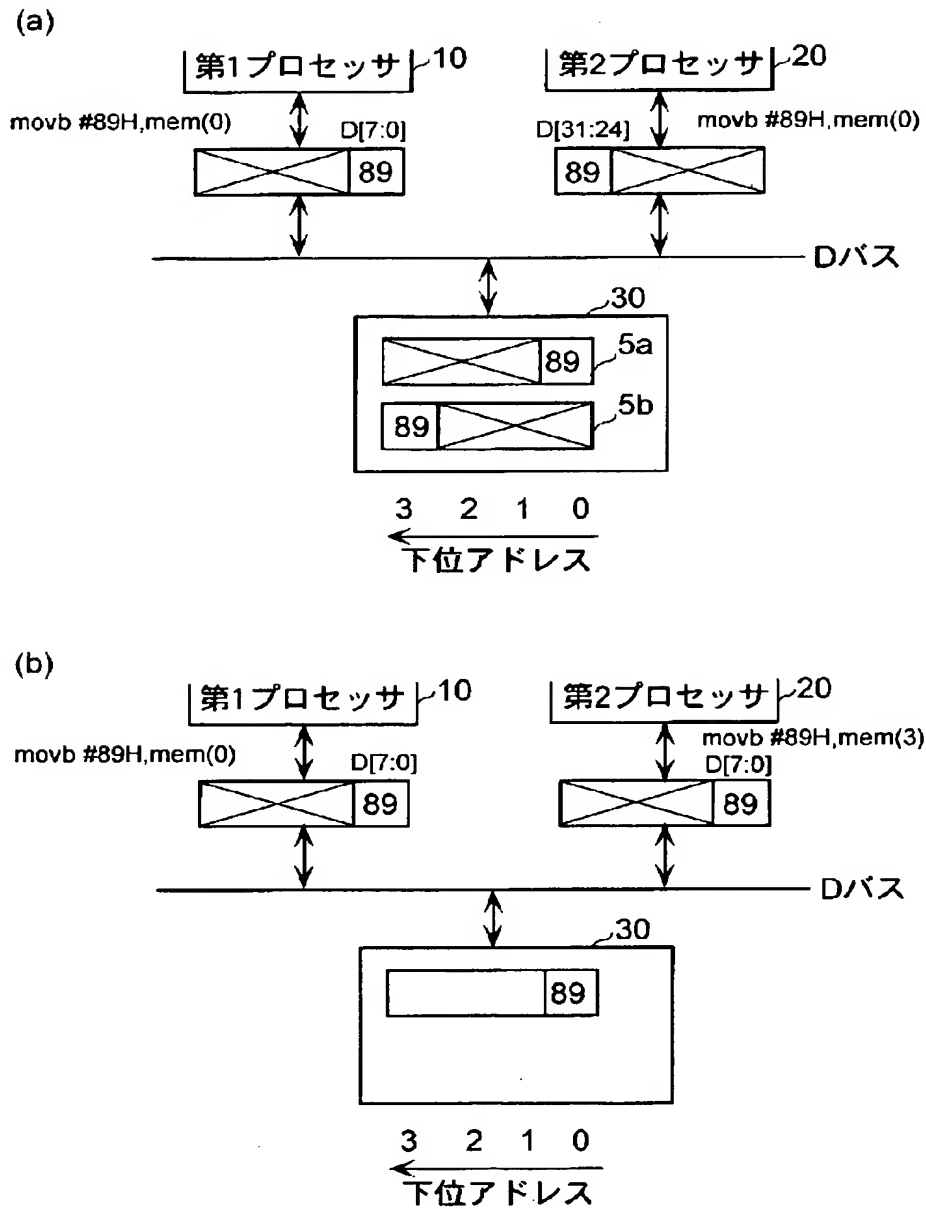
【図 3】



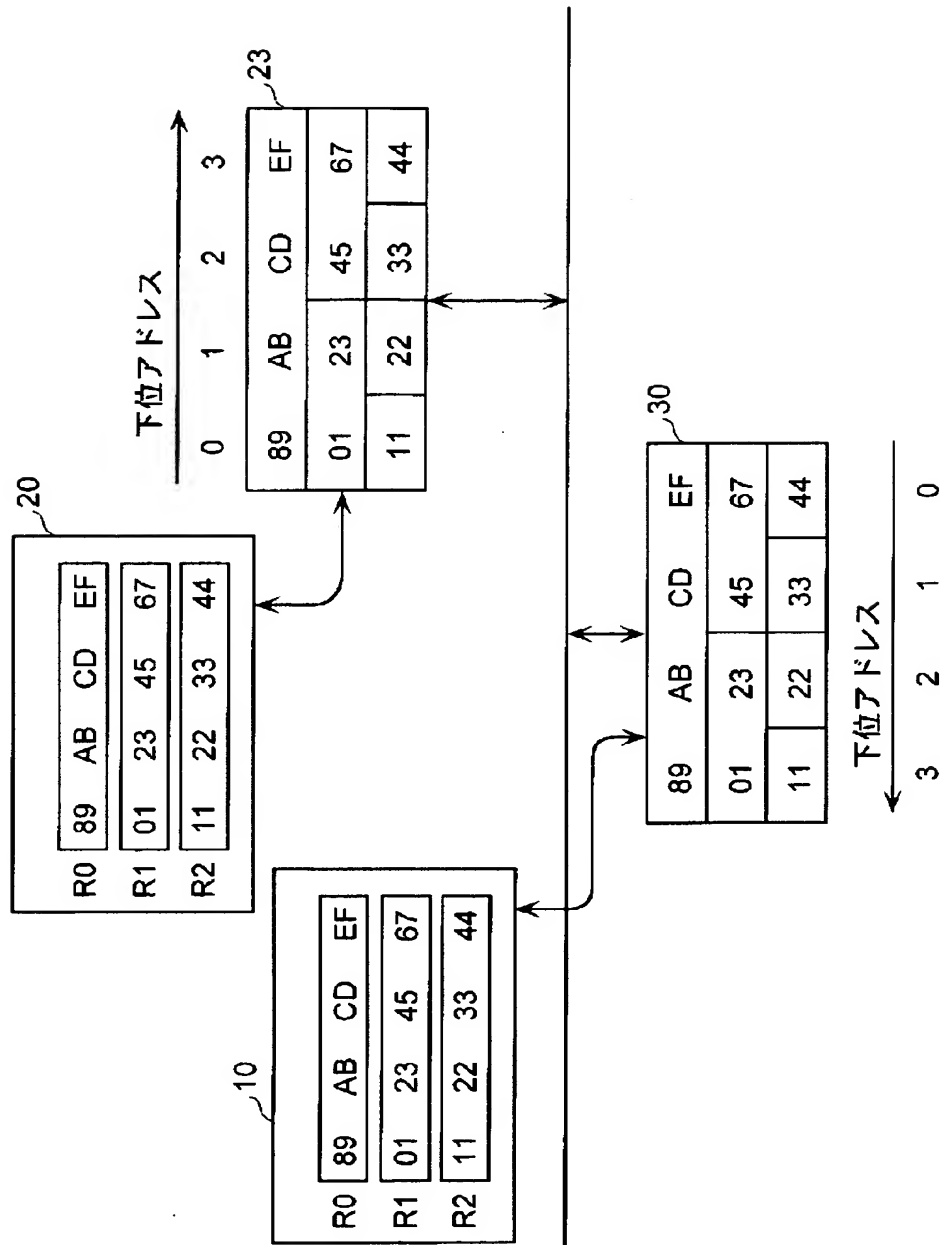
【図 4】



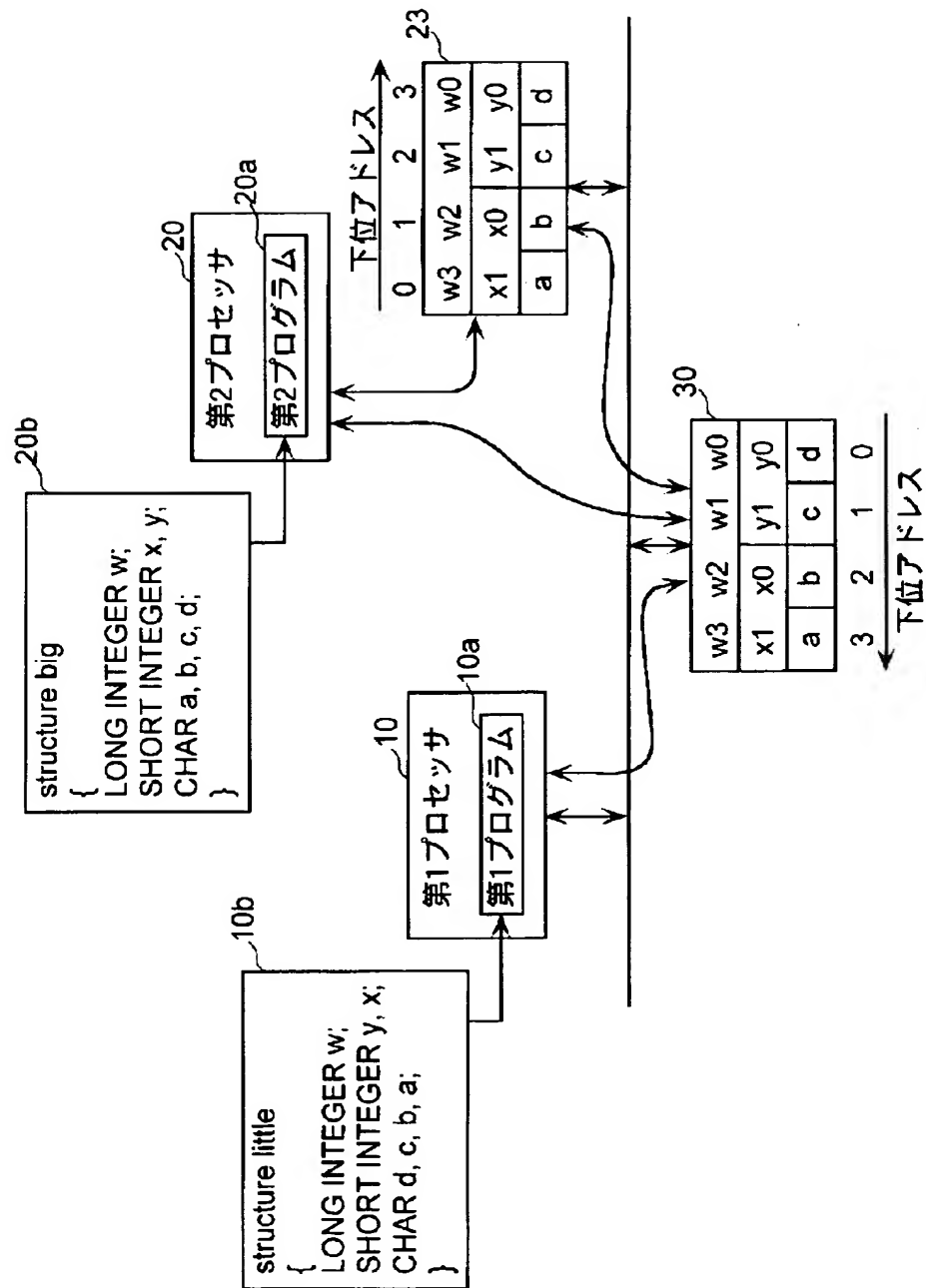
【図 5】



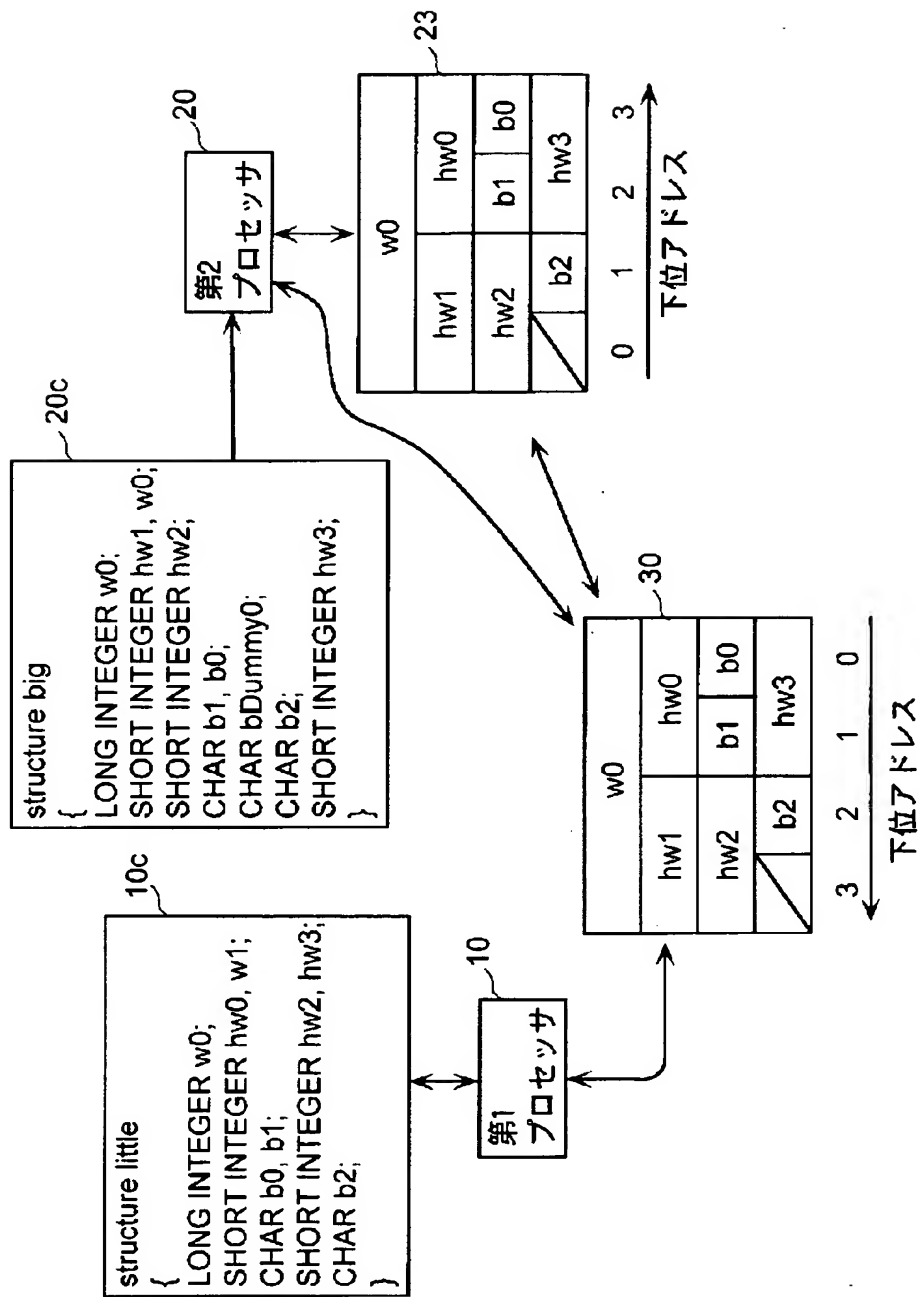
【図 6】



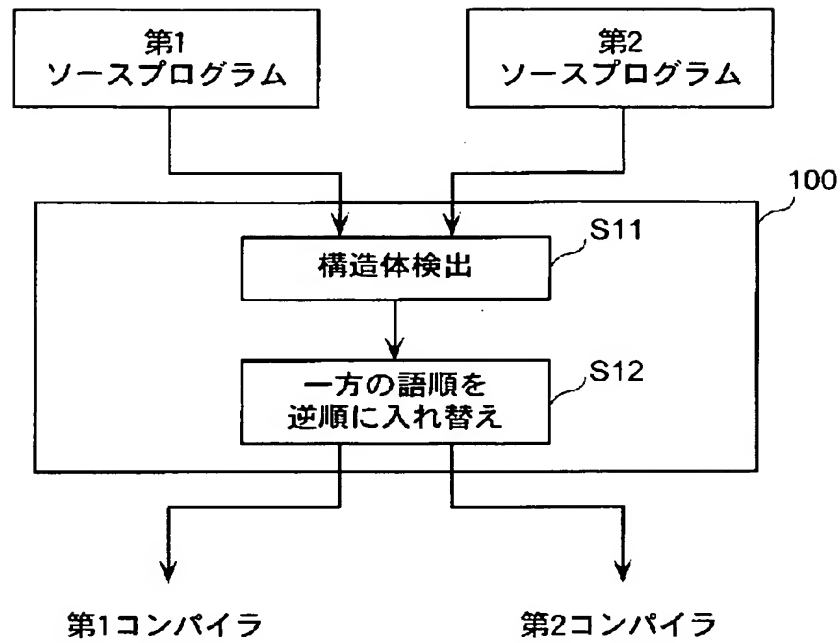
【図 7】



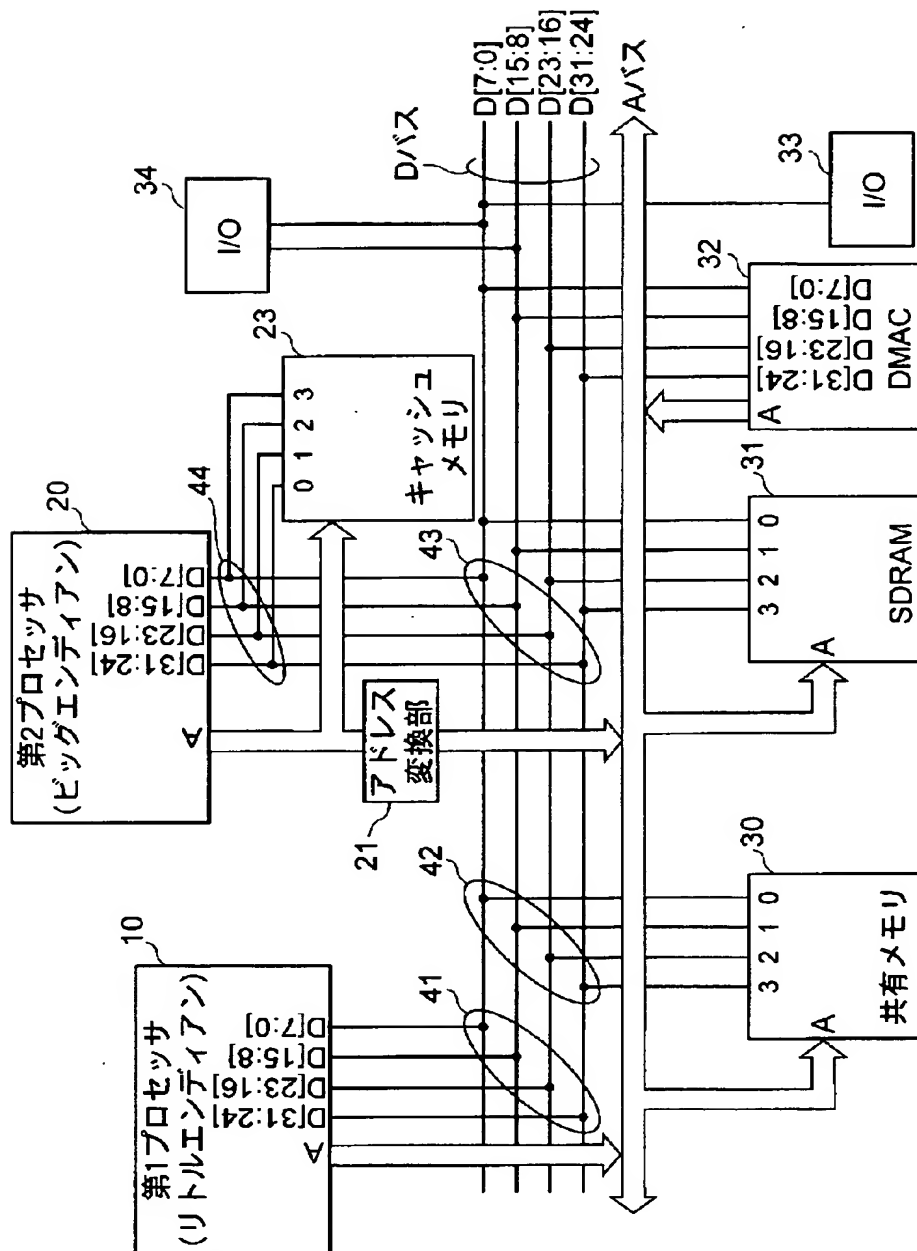
【図 8】



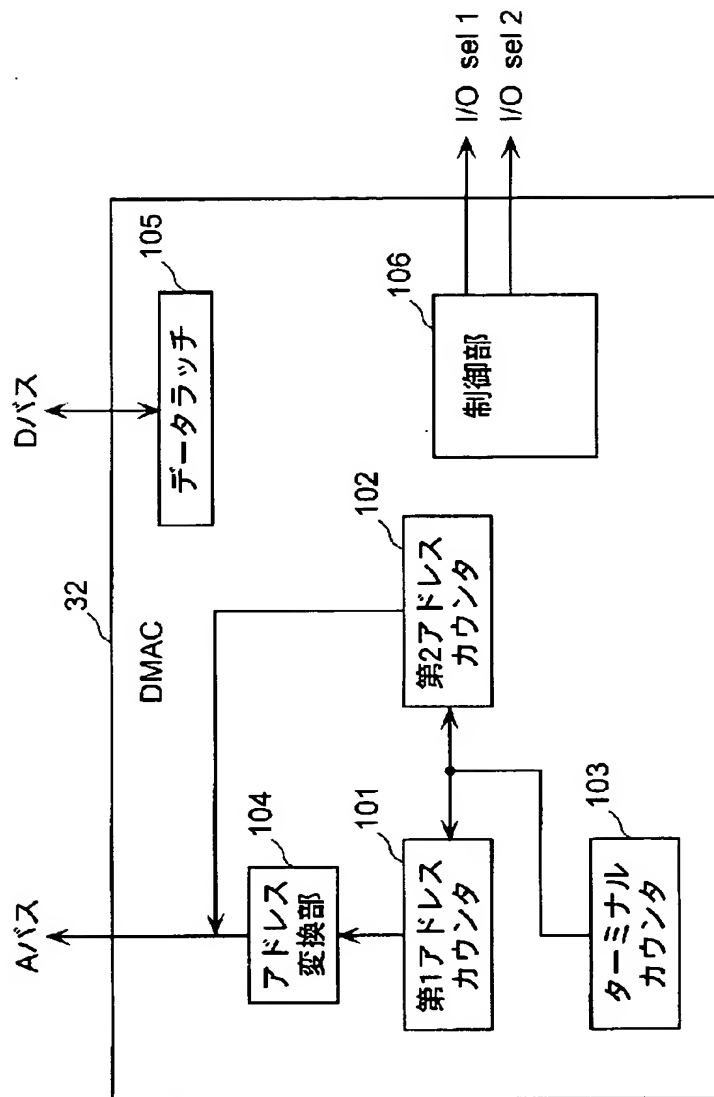
【図 9】



【図10】



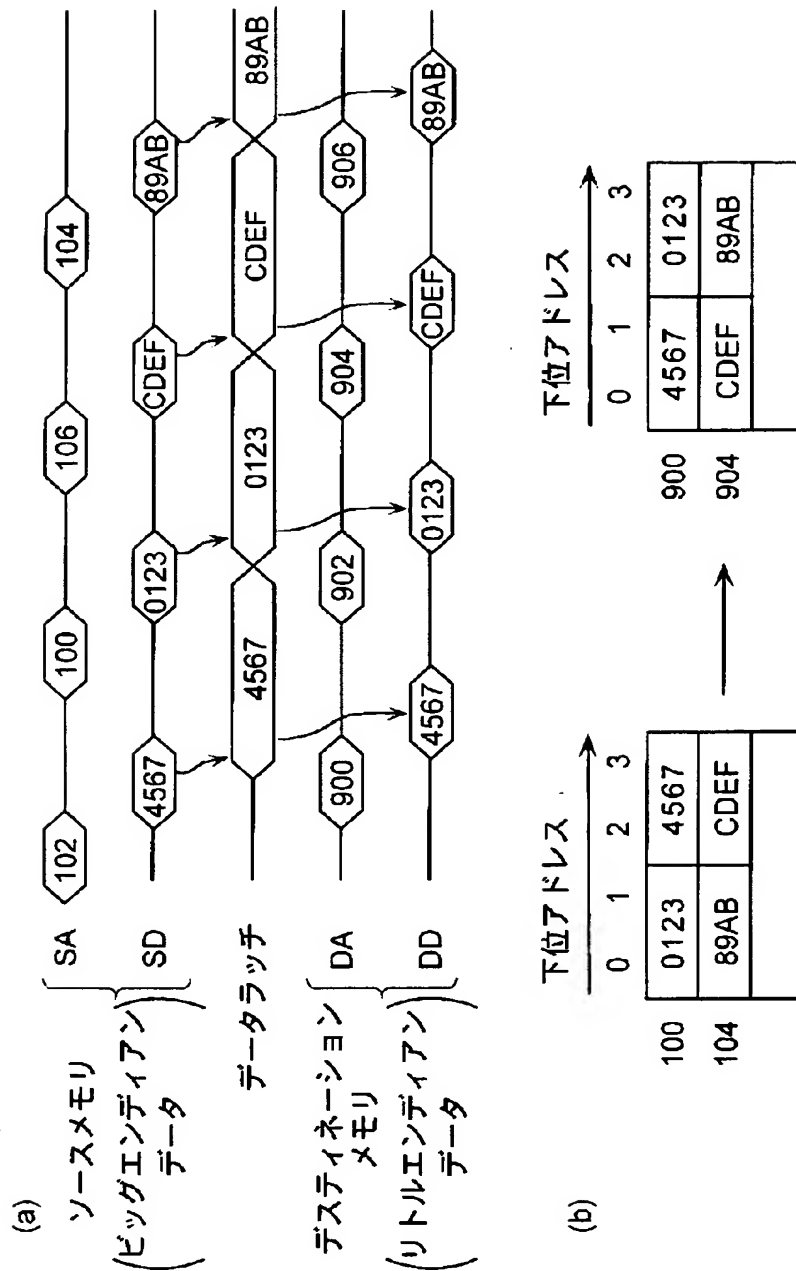
【図 11】



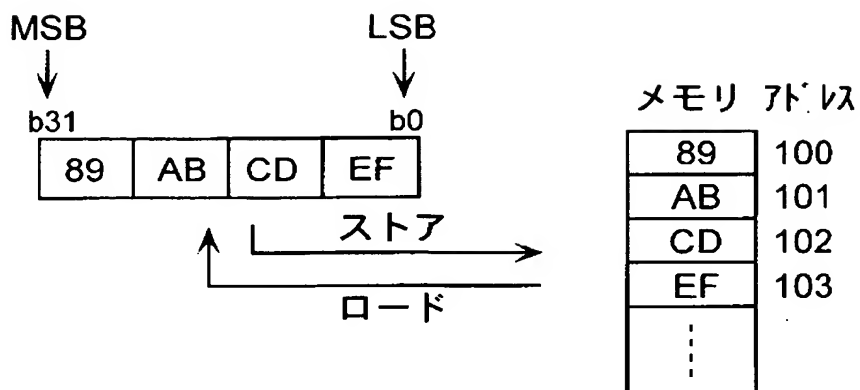
【図 12】

入力			出力
エンディアン	アクセスサイズ	変換前 アドレス	変換後 アドレス
同じ	×	A1, A0	A1, A0
異なる	w (32ビット)	A1, A0	A1, A0
異なる	hw (16ビット)	A1, A0	$\overline{A1}$, A0
異なる	b (8ビット)	A1, A0	$\overline{A1}$, $\overline{A0}$

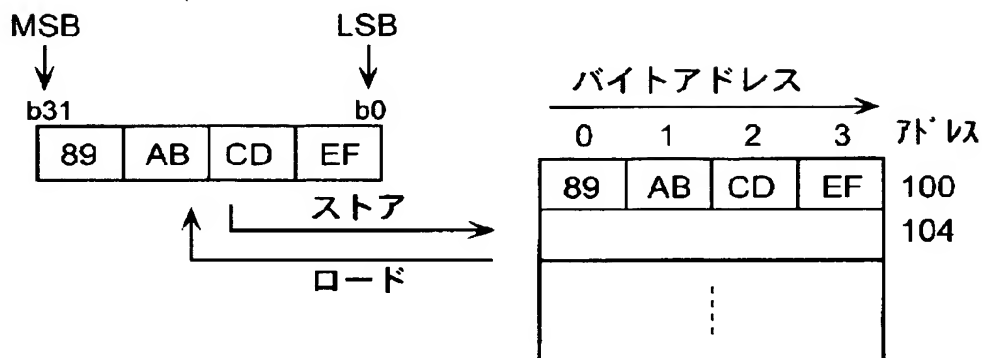
【図 13】



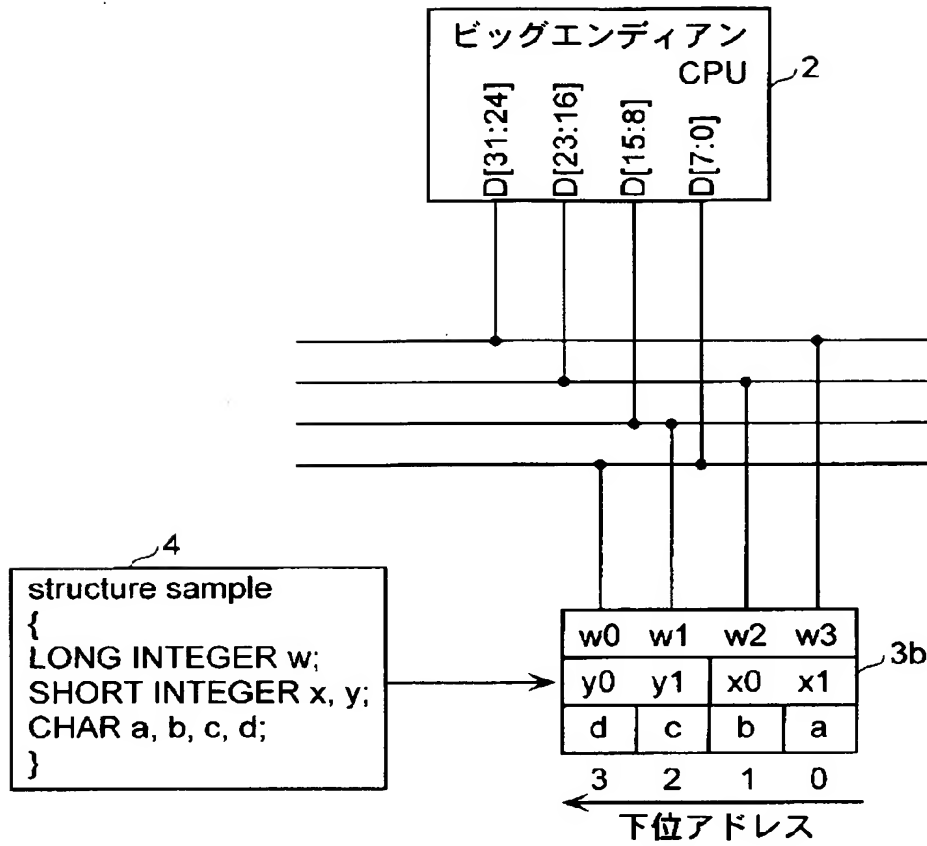
【図 14】



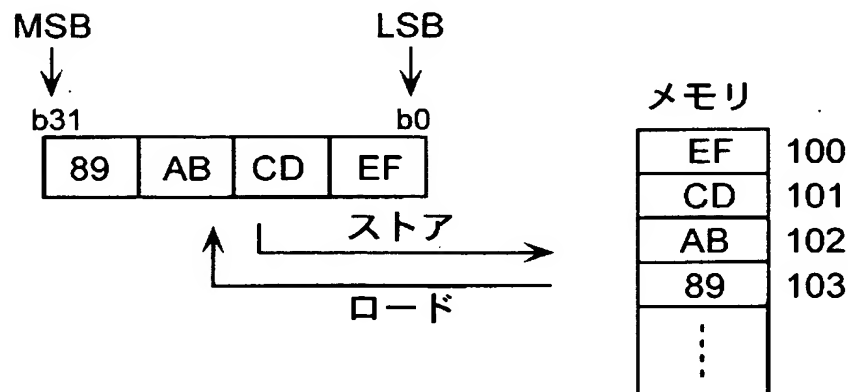
【図 15】



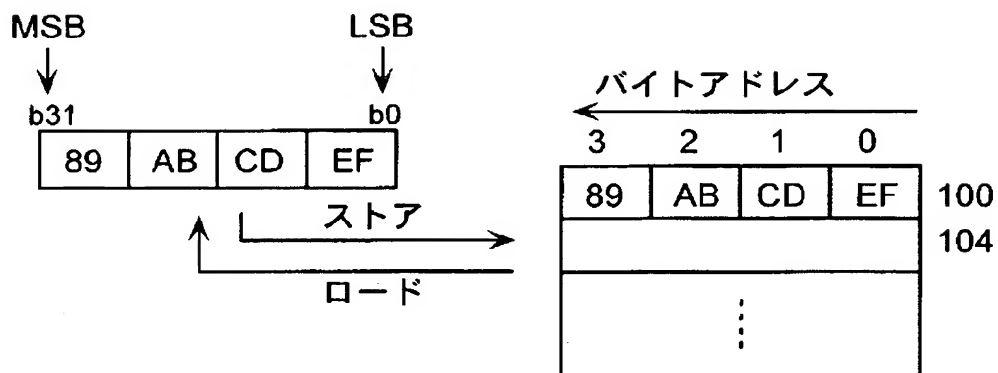
【図 16】



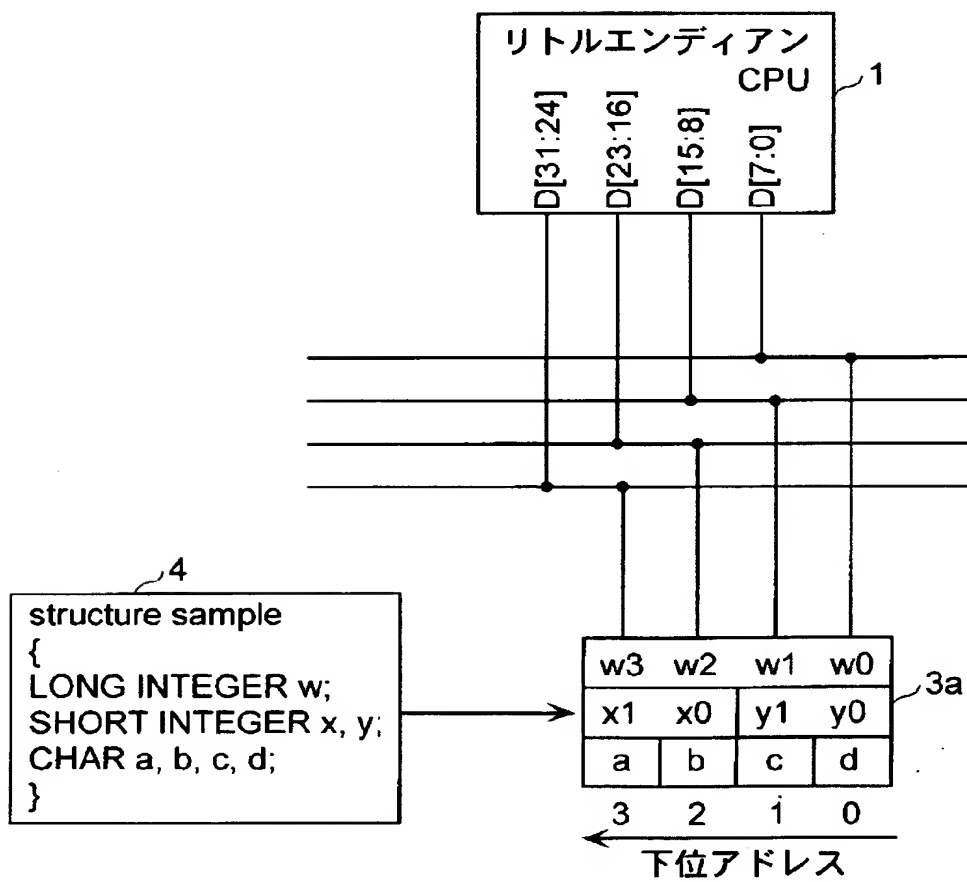
【図 17】



【図 18】



【図 19】



【書類名】 要約書

【要約】

【課題】 エンディアン異なるプロセッサにおいて極めて簡易な構成でメモリデータの共有を実現するメモリ共有装置を提供する。

【解決手段】 エンディアン異なる第1プロセッサ10及び第2プロセッサ20は、共に第1プロセッサ10のエンディアンによるバイトオーダでデータバスを介して前記メモリに接続され、第2プロセッサ20がデータバスよりも小さい幅のデータについて共有メモリにアクセスする場合に、データバスにおける当該データの位置を逆転させた位置を指すようにアドレスの下位ビットを変換して前記メモリに出力するアドレス変換部21を備える。

【選択図】 図1

認定・付加情報

特許出願の番号	特願 2 0 0 3 - 0 7 5 1 9 8
受付番号	5 0 3 0 0 4 4 7 6 7 2
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 5 年 3 月 2 0 日

< 認定情報・付加情報 >

【提出日】	平成15年 3月19日
-------	-------------

次頁無

特願 2 0 0 3 - 0 7 5 1 9 8

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 5 8 2 1]

1. 変更年月日	1 9 9 0 年 8 月 2 8 日
[変更理由]	新規登録
住 所	大阪府門真市大字門真 1 0 0 6 番地
氏 名	松下電器産業株式会社